AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A cache comprising:

a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of cache line in reverse program order, wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line.

2. (Previously Presented) The cache of claim 1, wherein the instruction segment is an extended block.

3. (Previously Presented) The cache of claim 1, wherein the instruction segment is a trace.

4. (Previously Presented) The cache of claim 1, wherein the instruction segment is a basic block.

5. (Previously Presented) A segment cache for a front-end system in a processor, comprising a plurality of cache entries to store instructions of instruction segments in reverse program order, wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal

instruction from the second instruction stream to be stored in a first position of a cache line.

6. (Previously Presented) The segment cache of claim 5, further comprising:

an instruction cache system,

an instruction segment system, comprising:

a fill unit provided in communication with the instruction cache system,

and

a selector coupled to an output of the instruction cache system and to an output of the segment cache.

7. (Previously Presented) The segment cache of claim 6, wherein the instruction segment system further comprises a segment predictor provided in communication with the segment cache.

8. (Previously Presented) A method comprising:

building an instruction segment based on program flow, and

storing instructions of the instruction segment in a cache entry in reverse program order, wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line.

9. (Original) The method of claim 8, further comprising:

building a second instruction segment based on program flow, and

if the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment.

10. (Original) The method of claim 9, wherein the extending comprises storing the non-overlapping instructions in the cache in reverse program order in successive cache positions adjacent to the instructions from the first instruction segment.

11. (Original) The method of claim 8, wherein the instruction segment is an extended block.

12. (Original) The method of claim 8, wherein the instruction segment is a trace.

13. (Original) The method of claim 8, wherein the instruction segment is a basic block.

14. (Previously Presented) A processing engine, comprising:

a front end stage to build and store instruction segments, instructions provided therein in reverse program order, wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line, and

an execution unit in communication with the front end stage.

15. (Previously Presented) The processing engine of claim 14, wherein the front-end stage comprises:

an instruction cache system,

an instruction segment system, comprising:

a fill unit provided in communication with the instruction cache system,

a segment cache, and

a selector coupled to an output of the instruction cache system and to an output of the segment cache.

16. (Previously Presented) The processing engine of claim 15, wherein the instruction segments are extended blocks.

17. (Previously Presented) The processing engine of claim 15, wherein the instruction segments are traces.

18. (Previously Presented) The processing engine of claim 15, wherein the instruction segments are basic blocks.

19. (Previously Presented) The processing engine of claim 15, wherein the instruction segment cache system further comprises a segment predictor provided in communication with the segment cache.